Preprint
UCRL-JC-146951

Storage Area Networks and the High Performance Storage System

Harry Hulen and Otis Graf
IBM Global Services
1810 Space Park Drive
Houston TX 77058
Hulen: 281-488-2473, hulen@us.ibm.com
Graf: 281-335-4061, ofgraf@us.ibm.com
FAX 281-335-4231

Keith Fitzgerald and Richard W. Watson
Lawrence Livermore National Laboratory
7000 East Ave.
Livermore, CA 94550-9234
Fitzgerald: 925-422-6616, kfitz@llnl.gov
Watson: 925-422-9216, dwatson@llnl.gov

This article was submitted to
Nineteenth IEEE Symposium on Mass Storage Systems
April 15-18, 2002
College Park, Maryland

February 4, 2002

Storage Area Networks and the High Performance Storage System

Harry Hulen and Otis Graf
IBM Global Services
1810 Space Park Drive
Houston TX 77058
Hulen: 281-488-2473, hulen@us.ibm.com
Graf: 281-335-4061, ofgraf@us.ibm.com
FAX 281-335-4231

Keith Fitzgerald and Richard W. Watson
Lawrence Livermore National Laboratory
7000 East Ave.
Livermore, CA 94550-9234
Fitzgerald: 925-422-6616, kfitz@llnl.gov
Watson: 925-422-9216, dwatson@llnl.gov

Abstract
The High Performance Storage System (HPSS) is a mature Hierarchical Storage Management (HSM) system that was developed around a network-centered architecture, with client access to storage provided through third-party controls. Because of this design, HPSS is able to leverage today's Storage Area Network (SAN) infrastructures to provide cost effective, large-scale storage systems and high performance global file access for clients. Key attributes of SAN file systems are found in HPSS today, and more complete SAN file system capabilities are being added. This paper traces the HPSS storage network architecture from the original implementation using HIPPI and IPI-3 technology, through today's local area network (LAN) capabilities, and to SAN file system capabilities now in development. At each stage, HPSS capabilities are compared with capabilities generally accepted today as characteristic of storage area networks and SAN file systems.

1. Introduction

Storage Area Network (SAN) technology has a bright future as measured by its growing market acceptance. Web information source allSAN.com [10] reports that:

> Within the mainframe arena, SANs already represent upwards of 25% of data center traffic. Outside of the mainframe area, SANs are expected to account for 25% of external disk storage and approximately 50% of multi-user tape storage by 2003

We believe that SAN technology will only reach its full potential when it can be used to provide secure sharing of data between heterogeneous client systems. To realize this potential requires appropriate storage system software and hardware architectures. The

ideas in this paper are independent of any particular SAN technology (e.g. Fibre Channel or iSCSI). One use for such a capability is a SAN-based global file system. A generic disk-based file system provides capabilities such as a naming mechanism, data location management, and access control. A global file system extends this capability to multiple independent operating systems by using specialized protocols, locking mechanisms, security mechanisms, and servers to provide device access. A SAN-based global file system is distinguished from other global file systems by the characteristic that client computers access storage devices directly, without moving data through a storage server.

The High Performance Storage System design and implementation are focused on hierarchical and archival storage services and therefore are not intended for use as a general-purpose file system. HPSS is nevertheless a file system, and specifically, a global file system. While any client applications (such as a physics code) can access HPSS devices with normal Unix-like calls to the HPSS client API library, in typical implementations these applications are data transfer applications that transfer data between HPSS files and the local file system. HPSS has a network-centered architecture that separates data movement and control functions and offers a secure, global file space with characteristics normally associated with both LAN-based and SAN-based architectures.

Figure 1 illustrates a typical deployment of HPSS. Note in particular the separation of control and data transfer networks (which may be physical or logical). This inherent separation of control and data helps enable HPSS to present a secure, scalable, global file system image to its users and leads naturally to full global SAN file system capabilities in the near future. The terms “Mover” and “Core Server” in Figure 1 are fairly descriptive of their function, but they are more fully described in Section 5.

This paper tracks the development of concepts and implementation for the separation of control and data functions in storage systems and the importance of these concepts for SAN file systems. These concepts are rooted in work that began over two decades ago [9] and prototyped a decade ago in the National Storage Laboratory (NSL) [3]. Lessons learned at the NSL led to the architecture of the High Performance Storage System (HPSS), which today supports a variety of high-speed data networks [4, 5]. HPSS is a collaborative development whose primary partners are IBM and the U.S. Department of Energy. This collaboration has been in existence for a decade, and HPSS development is ongoing. We discuss simple extensions to HPSS to exploit today’s SAN technology within large-scale HSM storage systems. We conclude with a section on lessons learned.

2. SAN Terminology

Several definitions of a Storage Area Network exist as related to common, shared repositories of data. The Storage Networking Industry Association (SNIA) online dictionary offers the following definition of Storage Area Network [1]:

1. A network whose primary purpose is the transfer of data between computer systems and storage elements and among storage elements. Abbreviated SAN. SAN consists of a communication infrastructure, which provides physical

connections, and a management layer, which organizes the connections, storage elements, and computer systems so that data transfer is secure and robust. The term SAN is usually (but not necessarily) identified with block I/O services rather than file access services.

2. A storage system consisting of storage elements, storage devices, computer systems, and/or appliances, plus all control software, communicating over a network.

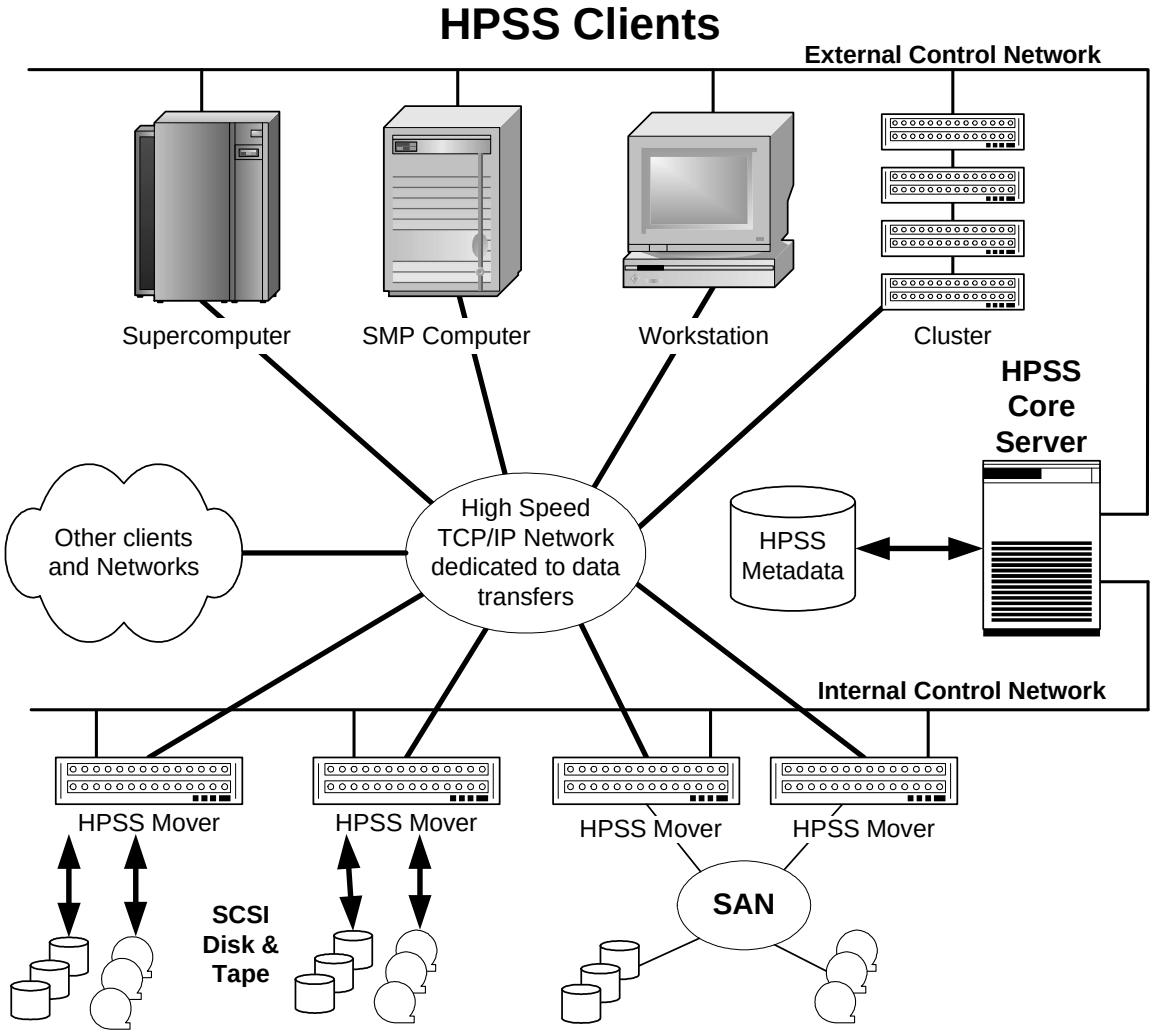

Figure 1: HPSS storage systems support a network centered architecture

Our interest is in large, high performance storage systems where 100s – 1000s of terabytes of data can be shared among client computers. The focus of SANs in our paper is from Bancroft et al [2]:

> The implementation [of a SAN] permits true data and/or file sharing among heterogeneous client computers. This differentiates [SAN file systems] from SAN systems that permit merely physical device sharing with data partitioned (zoned) into separate file systems. … The software orchestrating the architecture is what

unites the components and determines exactly how these elements behave as a system.

The same paper defines the notion of a SAN file system. Figure 2 illustrates the control and data flow of a such a generic SAN file system.

> The optimum vision is a single file system managing and granting access to data in the shared storage with high bandwidth Fibre Channel links [today there are other network technologies] facilitating transfers to and from storage. … The objective … *is to eliminate file servers* between clients and storage with minimum or no impact to the controlling applications. *Control information is typically separated from data traffic and in some architectures the two are isolated on completely separate networks.*

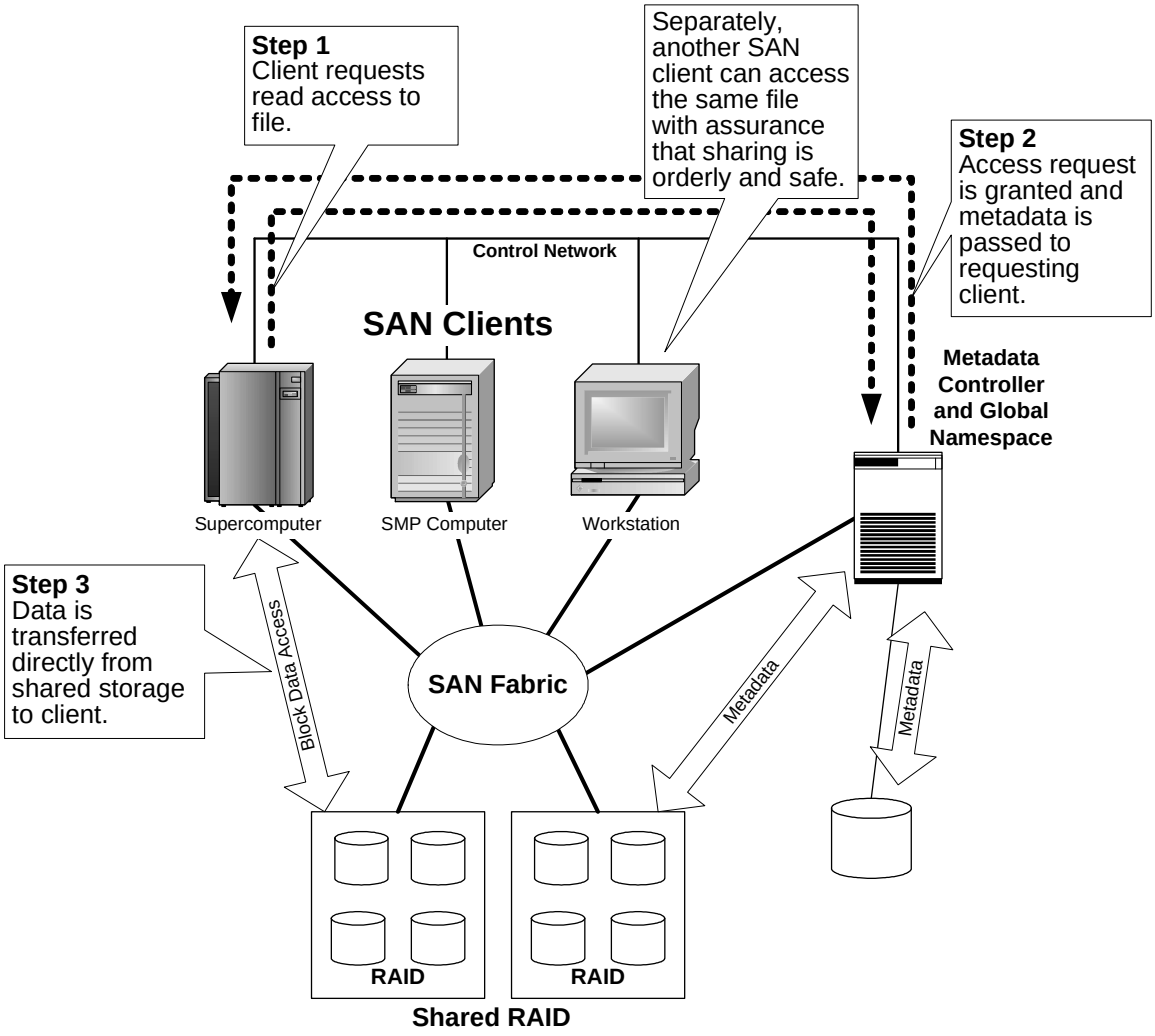

Figure 2: A file read operation illustrates the separation of data and control in a typical SAN file system.

It will be shown in the following sections that HPSS current implementation incorporates significant components of the SAN file system functionality described in the above definition, and how additional SAN file system functionality will be added to HPSS.

3. SAN Precursors

Although the term "SAN" is relatively new, the basic ideas of shared file systems have been around since the early days of computing. Papers by Thornton [8] and Watson [9] trace shared file concepts to the Octopus network at Lawrence Livermore National Laboratory in the 1960s, the Network Systems Corporation Hyperchannel, and the IEEE Mass Storage Reference Model in the late 1970s and early 1980s.

The foundation for HPSS can be traced to 1992 and the National Storage Laboratory (NSL). The NSL was a joint government/industry collaboration investigating high performance storage system architectures and concepts [3]. Work at the National Storage Lab led to NSL-Unitree, a prototype hierarchical storage system incorporating a distributed storage architecture that leveraged third-party data transfers almost a decade in advance of today's SAN deployments. A third-party data transfer is a data transfer controlled by an agent. The agent controls the data transfer by communicating with both the data source and the data sink in setting up the transfer. The agent does not participate in the actual movement of the data.

MAXSTRAT Corporation, a partner in the National Storage Lab, built high-end HIPPI-based RAID devices known as Gen4 and Gen5 arrays. These disk arrays were among the highest performing RAID disk devices of their day. Using the IPI-3 protocol, NSL-Unitree was able to achieve data rates of about 60 MB/s between a Cray C90 and MAXSTRAT disks over a HIPPI network.

IPI-3 was the third release of the Intelligent Peripheral Interface, a standards-based I/O interface that at the time was considered to be a high-end alternative to SCSI. Like SCSI, IPI-3 could exist as a native physical level protocol, or it could be encapsulated and sent over another general-purpose protocol such as HIPPI framing protocol. Disks were available equipped with a native IPI interface. Both IPI and TCP/IP could coexist on a HIPPI network through the use of HIPPI framing protocol.

The MAXSTRAT disk array was connected to a high performance computer via parallel or serial HIPPI, which has a nominal data rate of 100 megabytes per second. Originally designed as a point-to-point parallel interface, HIPPI evolved to be a switchable serial interface using a fibre transmission medium. Through the use of HIPPI switches, the Gen5 could be connected to multiple computers. By using encapsulated IPI, each computer could communicate with any Gen5 disk array as though it were a local IPI-3 device. Today this would be analogous to sharing a Fibre Channel disk array using SCSI over Fibre Channel, or more recently Gigabit Ethernet with SCSI over IP.

Significantly, the Gen4 and Gen5 implemented the third-party capabilities of the IPI-3 standard. With this capability, IPI-3 commands could be sent to a central server that mediated the requests and redirected them to source and sink for third-party transfer to

bring order and preserve data integrity. The following description of the third-party architecture from Chris Wood [6]:

> Third-party transfer architectures address the data "ownership" and access control issues by consolidating all data ownership and file system knowledge in a centralized server. Unlike NFS-style architectures, third-party transfer allows for direct disk I/O access to the central data store by clients. This architecture eliminates the burden of heavy inter-host lock manager and semaphore traffic and presents a well understood, NFS-like application interface. User data flows at local disk speeds (vs. network speeds) over dedicated high-speed disk channels while control traffic flows over a separate control network. The goal is to deliver data at optimal speeds with no interruptions for read/write commands and flow-control handshaking.

Essentially, The NSL proved the basic concepts of what we would now call a SAN file system. Figure 3 illustrates a file read operation in the NSL prototype. Note that Figure 3 is almost identical with Figure 2. Details of the protocol operation are given in [3].

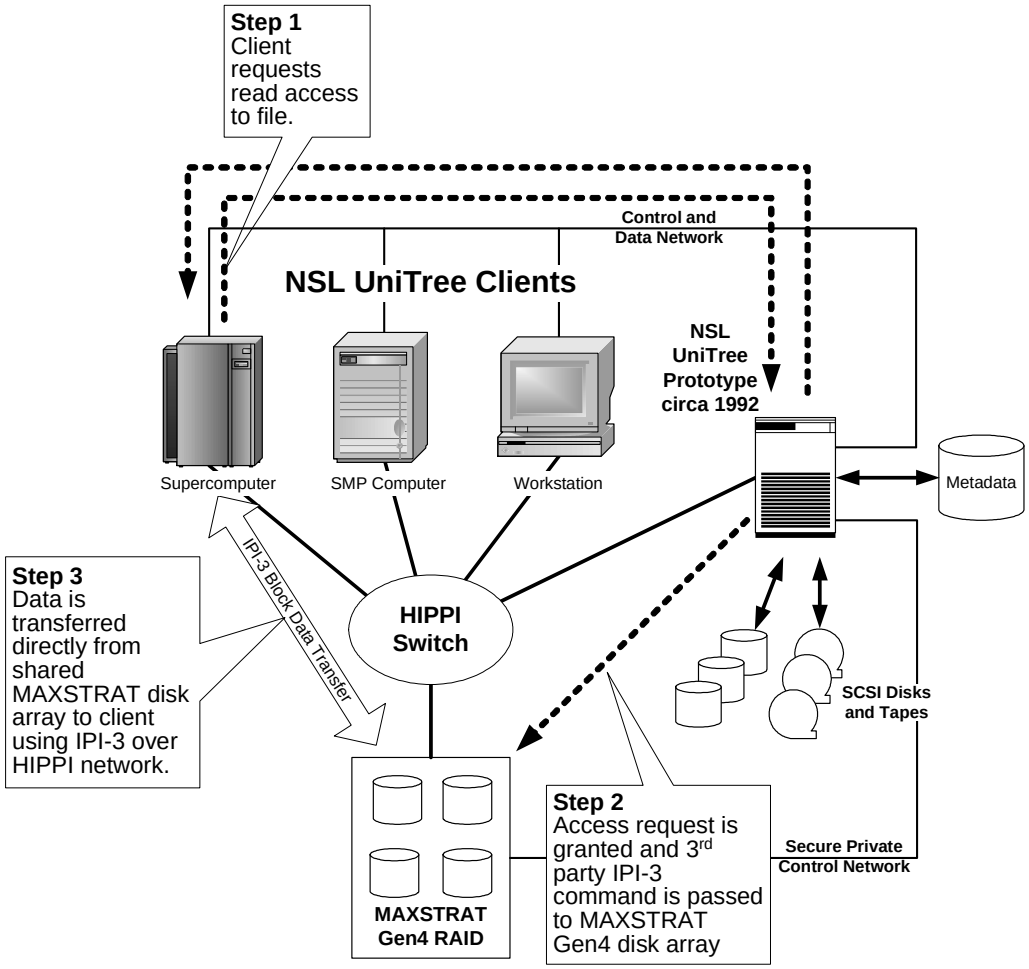

Figure 3: The NSL Prototype provided 3rd party "LAN-less" data transfers.

The NSL prototype proved several points to the NSL collaboration:

1. It established that data transfers between a client and network attached disks could give as good or better performance as native client disk.
2. Third-party data transfer allowed the transformation of the NSL server to function as a metadata engine that could effectively control data sharing among clients while maintaining high data rates.
3. Security is aided by separating control and data flow to separate networks.
4. Hierarchical storage, with movement of data between disk and tape, could be implemented in the shared disk environment.

4. Security Implications for SAN File Systems

Whenever data is shared among multiple clients, effective security mechanisms must be provided. In the case of robust global storage systems, security has historically been enforced by the file or storage server that effectively isolates clients from storage devices. NFS v4, AFS, DFS, and HPSS are examples of global storage systems that offer authenticated and authorized transfers between the client and storage servers. However when you make storage devices directly accessible to client systems, as in today's SANs, you have in effect opened a "Pandora's box" of security problems.

In today's SAN environments, shared storage appears as directly accessible devices on every client requiring access to the shared data. The level of protection for a shared SAN device is therefore no stronger than it would be for a local device attached to the client. This means that if any SAN client machine is compromised at the operating system root level, all shared data has been compromised. In effect, all shared-storage clients need to trust each other. SAN zoning limits visibility of devices to specified hosts and can be used to protect data by limiting access. But in cases where the goal is to make data globally accessible to many clients, security risks are incurred if any but the most trusted clients are added.

The NSL developers recognized this issue and provided a reasonable level of security by using a secure private control network connection between the storage servers and the network attached storage devices (See Figure 3). The storage system controlled access to all shared data. Clients did not have direct access to the storage devices because of the nature of the IPI-3 third-party protocol. Access to a network connection was granted to processes running on the storage clients on a per-transfer basis. The storage system used the secure private network to communicate with the MAXSTRAT disks, acting as the third-party agent facilitating all transfers between the storage clients and the network attached peripherals. It would have been very difficult for a rogue client to compromise the security of the NSL storage environment with this mechanism.

A similar level of security must be developed for use in a current SAN environment before the true power of SAN file systems can be safely realized. Object based "Network-Attached Secure Disks" [7] could solve this problem if they are accepted within the storage marketplace.

5. The Development of HPSS

The HPSS collaboration [4, 5] took up the work of the National Storage Laboratory collaboration in 1992 under a Cooperative Research and Development Agreement (CRADA) between IBM and several U.S. Department of Energy Laboratories (Lawrence Livermore, Los Alamos, Oak Ridge, and Sandia). After reviewing the projected requirements of next generation high performance HSM systems and all available hierarchical storage systems then in existence, the collaboration concluded that it was necessary to develop new software that would provide a highly scalable storage system, anticipating the growth in data-intensive computing (100s – 1000s of terabytes and Gigabyte/sec data transfer rate ranges) while also providing robust security for global file access. As this was to be a collaborative development, there was need for open access to source code among all collaboration members. The first production release of HPSS was in 1995, with major releases since then at approximately one-year intervals. Development is ongoing, with about 28 full time equivalent developers, including about 16 in the Department of Energy labs. Ongoing development is discussed in later sections. There are currently over 40 production HPSS sites worldwide in government, research, and education.

The scalability requirement led to a network-centered architecture that allowed scalability of storage capacity and data rates by adding management and storage elements to a scalable network. Like the earlier NSL prototype, HPSS was designed to accommodate intelligent third-party devices based on the model of the MAXSTRAT Gen4 and Gen5 disk arrays [4]. It was assumed that more intelligent third-party devices would follow; however, it was recognized that most of the storage devices that would be attached to HPSS would be conventional disks, disk arrays, and tape libraries. To accommodate conventional devices, the HPSS collaboration introduced the idea of a “Mover”. The notion was to attach SCSI disks and tape drives to low-cost computers running a lightweight HPSS Mover protocol. A data Mover and the disks and tapes attached to it formed the equivalent of an intelligent third-party device. Thus the HPSS architecture enabled both ordinary and intelligent devices and reasonably priced computers to work together while preserving security and a global name space.

Figure 4 illustrates the network-centered data flow of HPSS for a file read operation. Comparing this figure with the previous NSL illustration (Figure 3), one can see that the Mover and the disks and tape drives attached to it take on the attributes of an intelligent third-party device.

The HPSS Core Server presents the image of a file system to the user. Its main function is to manage the client interface and the system’s metadata (e.g. data location and security data). At the lower level involved with data transfer, the lightweight HPSS Mover code works only with block I/O. Unlike conventional network-attached storage (NAS), HPSS Movers transfer data over the network at a block level, not a file level, simulating the low-level I/O of early intelligent third-party devices and today’s SAN-attached devices. The Mover is strictly an intermediary to transfer logical blocks of data under control of the HPSS Core Server. See references [3, 4, 5] for details.

Use of multiple Movers allow many concurrent data transfers to provide very high aggregate data transfer rates. HPSS also supports data striping (parallel data transfers), thereby providing very fast single file transfer rates [4].

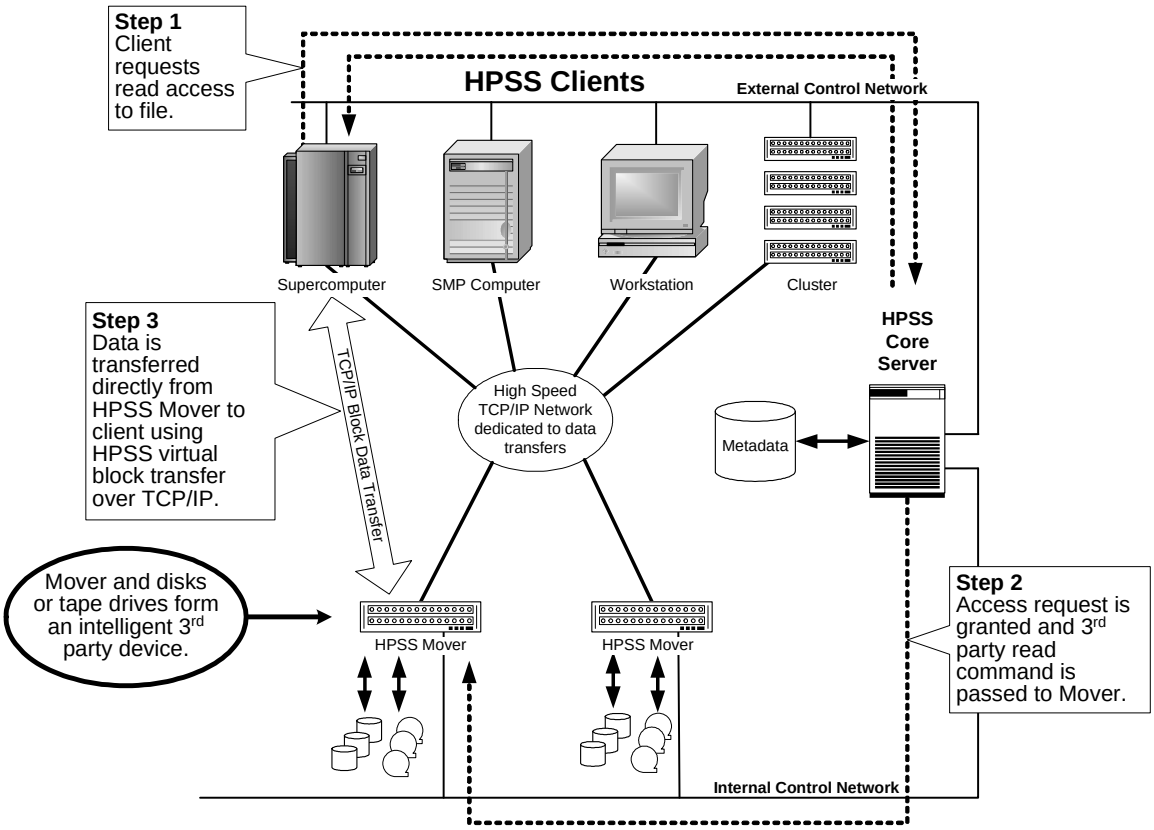

Figure 4: HPSS Movers create third-party capability using conventional devices.

HPSS, with its network-centered, third-party architecture is well suited to leverage SAN technology. The next section explains how SAN technology is used with HPSS today, and the sections that follow show enhancements will further exploit SAN technology.

6. Today's SANs and HPSS

Today's SAN technology promises better management and sharing of storage devices across HPSS Movers. SAN technology can simplify administration of large amounts of storage and can lead to better system reliability.

HPSS LAN-based configurations (refer back to Figure 1) are capable of providing very high bandwidths, both for individual data transfers and in the aggregate across concurrent file transfers and can furthermore support parallel, striped data transfers across multiple disks or tape drives. The current HPSS Mover architecture allows devices to be run at data transfer rates equal to 85% to 95% of the best possible device data transfer rates achievable at the block I/O level. Inexpensive network technologies such as Gigabit Ethernet, together with more efficient TCP/IP protocol implementations assure that LAN-centered technology is neither a performance bottleneck nor a cost issue for today's HPSS sites. Moore's Law has made Mover hardware inexpensive for lower I/O rate

devices such as tapes but for high throughput disk environments (100s MB/s per Mover) Movers are still relatively expensive. Thus, neither initial cost nor performance are sole motivators for introducing SAN technology into HPSS in some environments. For those requiring Movers capable of highest I/O rates, cost may be a motivator. SAN capabilities are important to the HPSS community because they will allow users of HPSS much more flexibility to reconfigure disks and tape drives when needs change.

The ability to reconfigure is especially important in case of component failures, including network, Mover, and device components. With SAN technology, disks and tape drives can be quickly reallocated among Movers, allowing quick restoration of service. Going one step further, SAN technology enables disks and tape drives to be connected to pairs of HPSS Movers, allowing the use of fault-tolerant software such as IBM's High Availability Cluster Multi-Processing (HACMP). All of these capabilities are available with today's HPSS just as they are available with other storage software, because SAN technology presents computers with the image of local disks or tape drives. Our goal is to exploit SAN technology as the high performance network connecting both clients and devices. This allows clients direct access to SAN devices, saving network store and forwards and data copies. Above the SAN level of device sharing and reconfiguration, HPSS adds the capabilities of a hierarchical, shared file system.

Having looked at how HPSS sites use SAN technology today to aid system administration and recovery from component failures, we now show how SAN capabilities can be expanded in future releases of HPSS.

7. SAN-enabled Movers and Clients

We have set a course to enable client applications to read and write data directly over a SAN, bypassing the existing store and forward character of TCP/IP networks when used with SCSI devices. In doing so, we will also enable HPSS to read and write data directly over a SAN for internal purposes such as migration and staging. The changes create "SAN-enabled Movers" and "SAN-enabled Clients."

We are currently evaluating a prototype that is an extension of the IPI-3 I/O redirection mechanism for disk access described earlier in the paper. Devices are assigned to a single Mover as is currently done in HPSS. In the case of I/O between a SAN-attached disk device and a SAN-attached client, the SAN-enabled disk Mover redirects its I/O descriptor (an internal HPSS data structure) to the client, which in turn can perform the I/O operation directly with the SAN disk. The "client" in this case could be either a true HPSS Client (i.e. a user application) or another Mover such as a tape Mover. No data passes through the disk Mover, as it is only used for the redirection control. Only a single disk Mover or a small number of disk Movers would be required, reducing cost. This design is called "I/O Redirect Movers."

We are also studying a design that allows HPSS to dynamically map a device to the a Mover for a data transfer. This design is called "Multiple Dynamic Movers." Currently devices are administratively assigned to specific Movers. With Multiple Dynamic Mover capability it will be possible to configure SAN-enabled Movers and Clients that are

equivalent to the I/O Redirect Mover capability in data transfer functionality and offer dynamic device to Mover mapping, which may be useful for dynamic failure recovery and load balancing. In the case of Clients, this would be accomplished by combining a SAN-Enabled Mover with a conventional Client API library.

We will have a prototype of SAN-enabled Movers and Clients running in an HPSS testbed in the spring of 2002. Experience with that prototype and the other design and requirements studies under way will lead to our final implementation choices. The selection of the "I/O Redirect Mover" or the "Multiple Dynamic Mover" will be made by mid year 2002 so as to deliver a SAN-enabled product in 2003. The discussion that follows applies to either approach.

For most systems configured for SAN enablement, fewer Movers will be required. Data transfer across a LAN is avoided. However, SAN enablement of Movers and Clients will be optional, and existing LAN-based capabilities will be fully supported. Sites that elect to use SAN-enabled Movers and Clients will benefit from fewer "hops" between HPSS-managed disk and the user and between disk and tape. On the other hand, the stronger inherent security for shared storage that is afforded by the current HPSS Mover and LAN approaches will in general (independent of HPSS) motivate some sites to use SAN enablement only for HPSS internal functions of migration and staging, while retaining LAN-based client functions. This will be discussed in more detail in Section 10.

In the next two sections, we look at ways SAN-enabled Movers and Clients can be used.

8. LAN-less and Server-less Data Movement for HSM Stage and Migrate

The HSM stage/migrate function moves data between levels in the storage hierarchy, usually consisting of disk and tape. In the current HPSS architecture, each storage device is assigned to a single data Mover. Data that is being staged to disk or migrated to tape is transferred between the respective Mover machines over a high-speed TCP/IP network.

SAN architecture is capable of making storage devices directly accessible to all Mover platforms connected to the SAN. With SAN-enabled Mover approaches outlined above, one Mover computer (which may run multiple Mover processes) will have the I/O descriptors for both source and sink ends of the transfer. Thus it will have the capability to migrate data from disk to tape or to stage data from tape to disk without moving data across a LAN. Eliminating a LAN transfer should allow fewer Mover computers and fewer LAN data connections. This is shown in Figure 5.

Going one step further, when devices and clients are directly attached to a SAN, the potential exists for the actual data movement to take place without going through a Mover by using the SCSI third-party copy command from a third-party agent. This capability is used in some tape backup systems today, and the same capabilities can be applied to hierarchical storage. Since the HPSS Mover software in Figure 5 has the addresses of both the disk and tape drive (source and sink), it can be extended to provide this third-party SCSI copy service or use another SAN agent specializing in this service.

We expect to consider this Server-less data transfer capability in the near future and see it as a logical extension to the LAN-less SAN enablement described above.

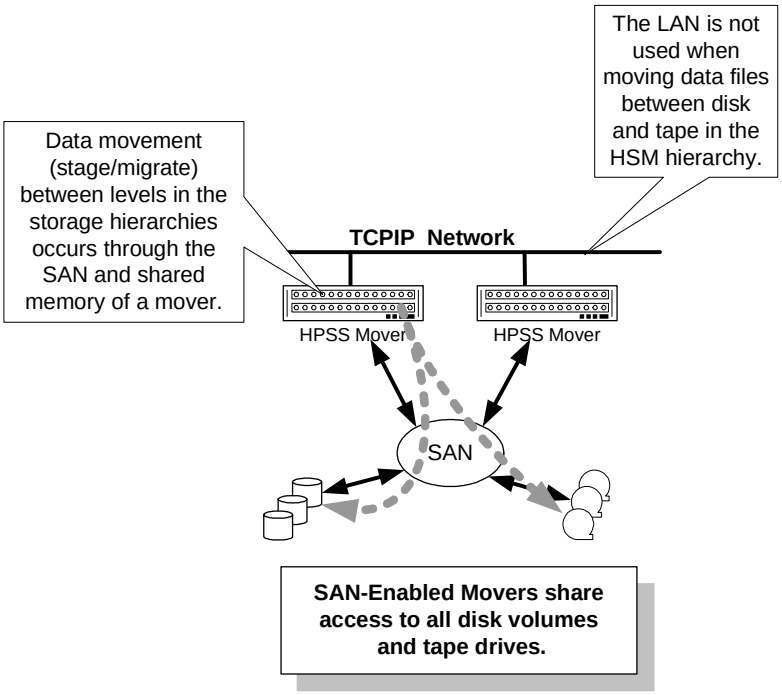

Figure 5: LAN-less Stage/migration between disk and tape using SAN-enabled Movers

9. LAN-less Data Movement between Clients and HPSS Storage Devices

The high performance user interfaces of HPSS are the Client API library, which is a superset of the Unix standard I/O read and write services augmented for parallel I/O, and Parallel FTP (PFTP), which is similarly a superset of Unix ftp. The Client API library, has code to support the Mover protocol and communicates with HPSS Movers using TCP/IP if the client and Mover are on different machines, or by an internal transfer mechanism if they are in the same computer.

SAN-enabled HPSS Client API libraries (Clients) will be able to access SAN-attached HPSS disks directly, and potentially also SAN-attached tapes. This can be done because the Client will be passed an I/O descriptor that describes the I/O operation to be performed. This is shown in Figure 6. The benefit of a SAN-enabled Client API library on a client machine must be weighed against the security exposure. This is discussed in the next section.

10. Security Considerations for Access to Storage: SAN versus LAN

We will now revisit security issues. Our assumption is that with today's generally available Unix-based technologies, a person who acquires root access, whether with authorization or not, can read and write any disk or tape that is configured as a local device. This includes SAN-attached devices. This is a well-known vulnerability of SANs, and it is the basic reason for zoning. The problem is that zoning and sharing data are

inherently at odds with each other. In an environment where access to a computer cannot be limited by physical means, then the information on shared devices is vulnerable to a rogue user with root access on any SAN-attached machine zoned for access to the shared data. (Zoning is a SAN capability that allows users to create multiple logical subsets of devices within a physical SAN as mentioned earlier. Access to devices within that zone is restricted to the members of the zone.) For this reason and until improved technology such as secure object-based devices [7] are available, server-facilitated access is currently the safest course for a file or storage system shared across computers.

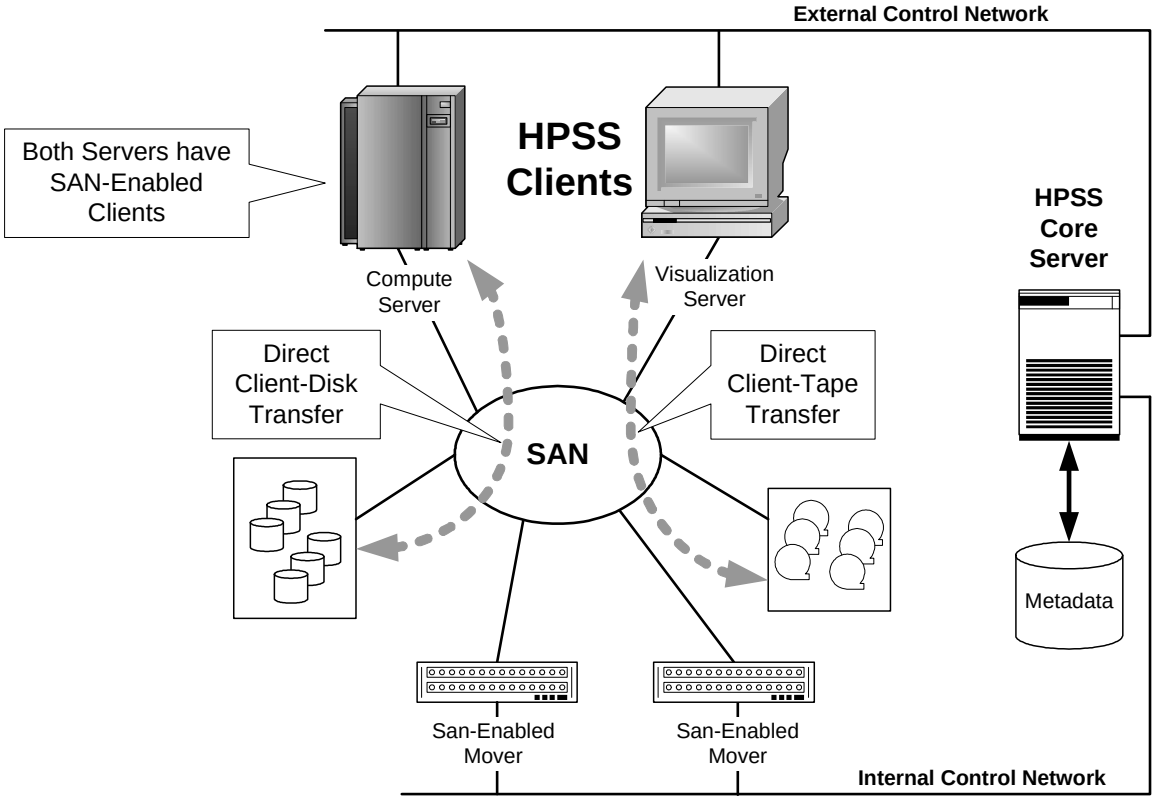

Figure 6: With SAN-enabled Movers and Clients, HPSS has LAN-less access to disk and/or tape storage.

Most large computer centers have systems with limited access that are not likely to be compromised. For systems where access can be limited and trust exists, sharing files across computers using SAN devices may present an acceptable level of risk.

Figure 7 shows appropriate use of current SAN and LAN capabilities for an example limited-access computer system and for an example open-access computer system. The configurations shown are typical of large IBM SP computers, large Linux clusters, and similar large-scale distributed architectures. By “limited access” we mean a computer system where access is physically controlled such that rogue users are very unlikely to gain access to the I/O client nodes, while an “open” system would be less secure and the I/O client nodes would be more vulnerable. For simplicity only the data paths are shown in Figure 7. Control would typically be over a fast Ethernet.

Each computer system in the example of Figure 7 has a local file system such as the IBM General Parallel File System (GPFS). GPFS is the principal file system for the IBM SP and is also used with Linux clusters. GPFS as configured here would provide access to files across nodes within each computer system but not across computer systems. Therefore GPFS data accessible to one system would be on disk zones not visible to the other computer system. This is the classic use of SAN zoning to protect each computer system's local file system. Use of SAN zoning to allocate storage to HPSS and local file systems is the heart of the administrative benefit of SANs.

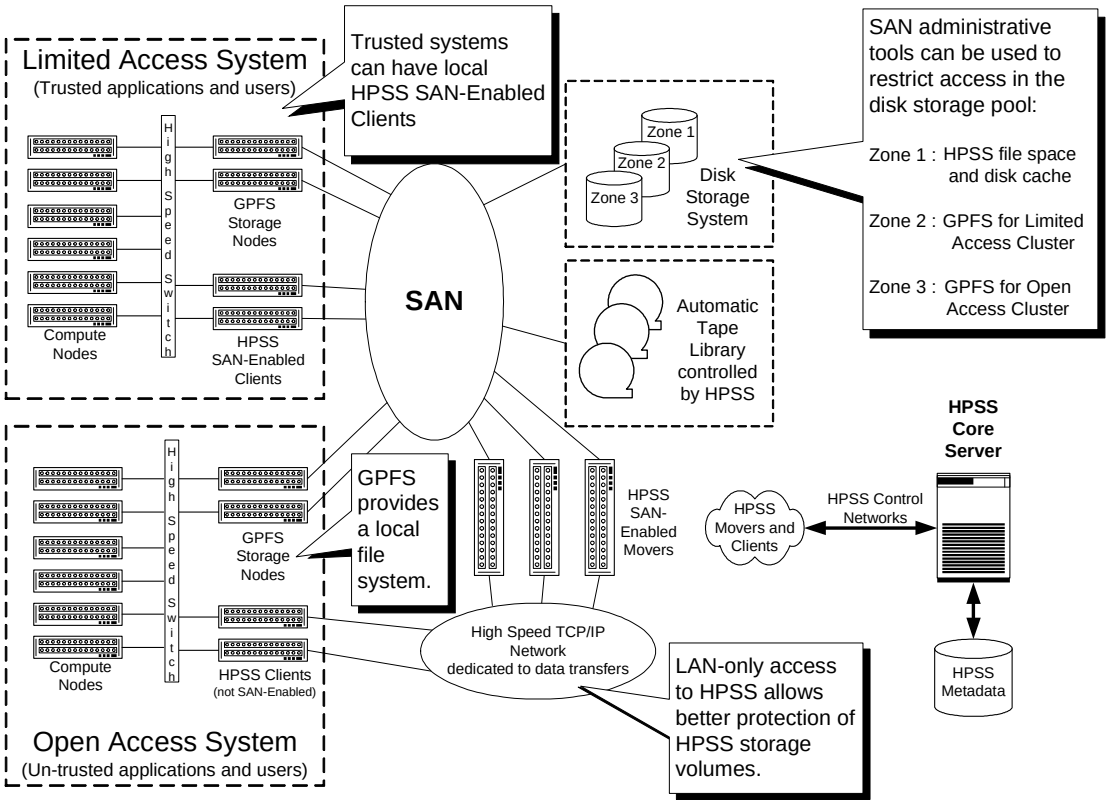

Figure 7: Example of where HPSS provides a global file system to both trusted and un-trusted clients.

HPSS, on the other hand, is typically configured such that files are globally visible across all HPSS client computers (although HPSS clients can be configured with limited access to particular classes of HPSS files). In the above example, all HPSS Client nodes in both clusters have access to all the files. The SAN zones are configured such that the SAN-enabled client nodes and SAN-enabled mover nodes all have access to the HPSS disk cache. As a result, data transfers from HPSS disks to nodes in the Limited Access System cluster will occur over the SAN and no external LAN is required for data transfer. SAN terminology would be "LAN-less" or "LAN-free" transfer.

For a cluster with a reasonably small number of compute nodes, it would be possible to put a SAN-enabled Client on each compute node, thereby eliminating the need to transfer data across the backbone network of the cluster. However for a large cluster or SP, this would require an equally large SAN switch. It would also open the HPSS data zones to

the previously described vulnerability of SANs to rogue users with root access to the compute nodes. This vulnerability is not a limitation of HPSS but is due to the lack of security mechanisms to protect shared data in today's SANs. It would therefore be recommended that in most situations, dedicated nodes be used for the HPSS Clients. At LLNL, for example, a utility that provides persistence and queuing capabilities is frequently utilized when moving data between a host's local file system (e.g. GPFS above) and HPSS storage. This utility's non-interactive queue based structure allows it to execute on a protected I/O node that does not allow user logins.

The Open Access System, which is the less trusted of the two systems, is configured to access HPSS files only through the LAN, using conventional capabilities of the HPSS Client without SAN enablement. This provides the maximum protection for HPSS data.

11. Lessons Learned

The HPSS collaboration and the earlier NSL collaboration have dealt with the problems of scalable, network-centered storage for over a decade. Our charter is to provide storage software for large, demanding applications such as those of the Department of Energy labs that sponsor HPSS. Other large applications where HPSS has been deployed include supercomputer centers, weather, high-energy physics, and defense. Our "lessons learned" apply both to this high end of hierarchical storage and archiving and we believe to SAN file systems generally. Our experience has led us to a blend of LAN-based and SAN-based technologies with the overarching requirements of scalability, high data rates, shared access to files, security, high availability, and manageability.

Based on our experience with HPSS and our forty plus installations we have found that:

- High data rates and scalability are supported by a network-centered architecture, but not tied to either LAN or SAN.
- The lightweight HPSS Mover, which is based on a concept from the IEEE Mass Storage Reference Model Version 5, is a useful tool for scalability and facilitates simple evolution toward full support for SAN file system concepts.
- LAN-based and SAN-based technologies are complementary and can be mixed.
- Data rates are limited by the hardware configuration (including the network and the choice and number of devices) and not by HPSS software.
- Due to the lack of an adequate SAN security mechanism, shared access to data is best managed in a server-based environment for situations requiring protection from a rogue users who might obtain root access.
- Manageability and high availability are enhanced by SAN capabilities.
- Separation of data network paths from control network paths enhances security.

We find that the blending of LAN and SAN capabilities of current and future releases of HPSS effectively addresses scalability, high data rates, shared access to files, security, availability, and manageability ways that are useful to high-performance data-intensive computing. We believe that the lessons of NSL and HPSS have applicability to others in our industry exploring or developing SAN based file and storage systems, as the current explosion of electronic data goes on around us.

12. Acknowledgements

We wish to thank the early participants in the National Storage Laboratory for their support of early network centered storage architectures and the many developers within the HPSS Collaboration who have created HPSS. This work was, in part, performed by the Lawrence Livermore National Laboratory, Los Alamos National Laboratory, Oak Ridge National Laboratory, National Energy Research Supercomputer Center and Sandia National Laboratories under auspices of the U.S. Department of Energy, and by IBM Global Services - Federal.

References

[1] The Storage Network Industry Association (SNIA) has an excellent dictionary on their web site, www.snia.com. The dictionary is currently located in the "Resource Center" area of the web site. The definitions in this paper differ somewhat from SNIA definitions, but the authors acknowledge the authority of the SNIA dictionary.

[2] M. Bancroft, N. Bear, J. Finlayson, R. Hill, R. Isicoff, and H. Thompson, "Functionality and Performance Evaluation of File Systems for Storage Area Networks (SAN)," *Proceedings Eighth Goddard Conference on Mass Storage Systems*, College Park, MD (Mar 2000). This paper has an excellent overview of SAN file systems.

[3] R. Hyer, R. Ruef, and R. W. Watson, "High Performance Direct Network Data Transfers at the National Storage Laboratory," *Proceedings Twelfth IEEE Symposium on Mass Storage*, Monterey, CA (Apr. 1993). This paper documents the history of NSL-Unitree and 3rd party IPI-3.

[4] R. A. Coyne and R. W. Watson, "The Parallel I/O Architecture of the High Performance Storage System (HPSS)," *Proceedings Fourteenth IEEE Symposium on Mass Storage, Monterey*, CA (Sept. 1995)

[5] D. Teaff, R. W. Watson, and R. A. Coyne, "The Architecture of the High Performance Storage System (HPSS)," *Proceedings Goddard Conference on Mass Storage and Technologies*, College Park, MD (Mar. 1995). For more recent HPSS architectural information, refer to the HPSS web site www.clearlake.ibm.com/hpss.

[6] C. Wood, "It's Time for a SAN Reality Check," available at http://www.maxstrat.com/san_wht.html. This paper includes a discussion of third-party data transfer as implemented in the MAXSTRAT Gen4 and Gen5 disk arrays.

[7] Garth A. Gibson, David F. Nagle, Khalil Amiri, Fay W. Chang, Howard Gobioff, Erik Riedel, David Rochberg, and Jim Zelenka. "Filesystems for Network-Attached Secure Disks" CMU-CS-97-118 July 1997

[8] Thornton, James E., "Back-end Network Approaches", IEEE Computer, Vol. 13, No. 2, Feb. 1980, pp 10 -17. This paper reviews the history of storage network approaches and outlines the directly attached storage features of Hyperchannel.

[9] Watson, Richard W., "Network Architecture Design for Back-End Storage Networks", IEEE Computer, Vol. 13, No. 2, Feb. 1980, pp 32-49. This paper reviews why a shared file system approach is critical to success of storage networks and outlines the architecture that became the IEEE Reference Model, UniTree and HPSS, including third-party transfers, Movers, and direct device to device transfers.

[10] allSAN Research Services, http://www.allsan.com/marketresearch.php3